CMPG-767 Digital Image Processing

SPATIAL DOMAIN FILTERING
NONLINEAR FILTERS

Nonlinear Filtering

- As well as any linear filter, a nonlinear filter is defined by an operator

$$\hat{f}(x,y) = T\left(g(x,y)\right)$$

where $g(x,y)$ is an image to be processed, T is a filtering operator and $\hat{f}(x,y)$ is a resulting image

- Linearity does not hold for a nonlinear filter

$$T\left(af(x,y) + b\eta(x,y)\right) \neq aT\left(f(x,y)\right) + bT\left(\eta(x,y)\right)$$

Nonlinear Spatial Domain Filtering

- Nonlinear spatial filtering of an image of size M x N with a filter of size m x n is defined by the expression

$$\hat{f}(x,y) = T(S_{xy})$$

where S_{xy} is a local m x n window around the pixel $g(x,y)$ in the image g to be processed, and T is a nonlinear filtering operator

Variational Series – Series of Order Statistics

- An arrangement of the values of a random sample $x_1, \ldots, x_n$ with distribution function $F(x)$ in ascending sequence where $x_{(1)} \le x_{(2)} \le \ldots \le x_{(n)}$

$$x_{(1)} = \min_{i=1,\ldots,n} x_i; \quad x_{(n)} = \max_{i=1,\ldots,n} x_i$$

- The series is used to construct the empirical distribution function $F_n(x) = m_x / n$, where m_x is the number of terms of the series which are smaller than x.

Variational Series – Series of Order Statistics

- A rank (order statistics) of an element in a variational series is its serial number in the series

$$x_{(1)} \le x_{(2)} \le \ldots \le x_{(n)}$$

Rank 1 Rank 2 … Rank n

Variational Series – Series of Order Statistics

- Example1. Let us have the following sample
  200, 101, 102, 125, 5, 10, 207, 180, 100

  Its variational series is
  5, 10, 100, 101, 102, 180, 200, 201, 207
- Example2. Let us have the following sample
  200, 101, 102, 101, 207, 101, 100, 207, 5

  Its variational series is
  5, 100, 101, 101, 101, 102, 200, 207, 207

Median Filter – the Simplest Spatial Domain Nonlinear Filter

- Median filter replaces an intensity value in each pixel by a local median taken over a local n x m processing window:

$$\hat{f}(x,y) = \mathrm{MED}\left(S_{xy}\right)$$

where S_{xy} is a local m x n window around the pixel (x,y) in the image g (to be processed), and MED is the median value taken over the window S_{xy}

Median Filter – Implementation

- To implement the median filter with an n x m processing window, it is necessary to build a variational series from the elements of the processing window S_{xy} around the pixel with coordinates (x,y) (a pixel to be processed)

- The central element of the vatiational series is the median of the intensity values in the window S_{xy}

Median Filter and Impulse Noise

- Median filter is highly efficient for impulse noise filtering
- Median filter with 3x3 window can almost completely remove impulse noise with the corruption rate up to 30%
- Applied iteratively, it can remove even noise with a higher corruption rate

Median Filter and Impulse Noise

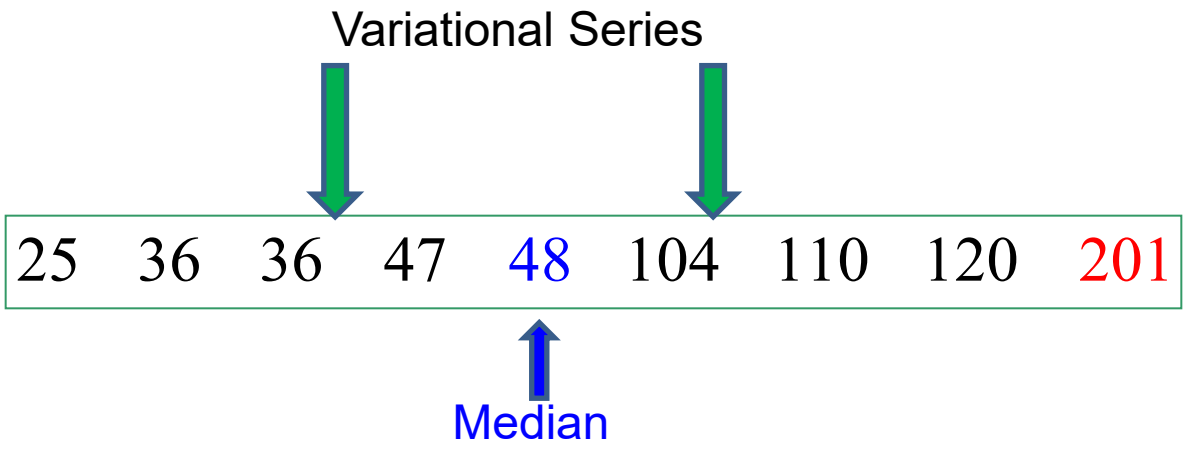
- The ability of the median filter to remove impulse noise is based on the following:
  - ➢ Impulses have unusually high (or low) intensity values compared to their neighbors

- This means that they are located in the left or right ends of the variational series built from the intensity values in a local window S_{xy} around the pixel with coordinates (x, y)

Median Filter and Impulse Noise

3x3 window, 201 is an impulse

25	36	36
47	201	48
104	110	120



The processing result
The impulse 201 was removed

25	36	36
47	48	48
104	110	120

Mean Filter and Impulse Noise

- Unlike the median filter, the mean filter is unable to remove impulse noise. It only “washes” impulses

3x3 window, 201 is an impulse

25	36	36
47	201	48
104	110	120

Mean = **81**

The processing result
The impulse 201 was just “washed”

25	36	36
47	81	48
104	110	120

Disadvantages of Median Filtering

- Median filter removes impulse noise, but it also smooths ("washes") all edges and boundaries and may "erase" all details whose size is about $n/2 \text{ x } m/2$,where $n \text{ x } m$ is a filter window size
- As a result, an image becomes "fuzzy"
- Median filter is not so efficient for additive Gaussian noise removal

Detection of Impulse Noise

- To reduce image smoothing by the median filter, impulse detectors should be used
- A detector analyses local statistical (and possibly other) characteristics in a local window around each pixel using some criteria, and marks those pixels that are corrupted by noise
- Then only marked pixels shall be processed by the median filter

Differential Rank Impulse Detector

(introduced by I. Aizenberg and C. Butakoff in "Effective Impulse Detectors Based on Rank-Order Criteria", *IEEE Signal Processing Letters*, Vol. 11, No 3, March, 2004, pp. 363-366

- This detector is based on the following reasonable assumptions

➢ Impulses are located either in the ends of the variational series (salt-end-pepper and bipolar noise) or close to these ends (random noise)

➢ The difference between the intensity values of impulse and its neighbor located in the variational series between this impulse and the **median** should exceed some reasonable threshold value

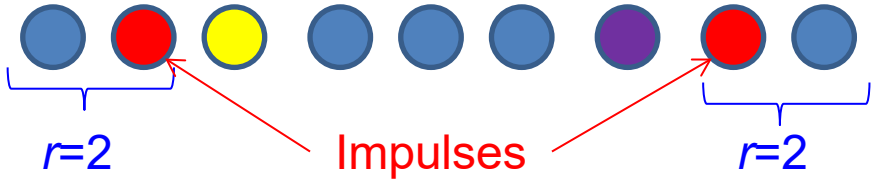
Differential Rank Impulse Detector

- Let $R(g(x,y))$ be the rank of an intensity in the pixel $g(x, y)$ in the variational series and $\mathrm{var}(k)$ be the intensity value of the pixel whose rank is k

$$d_{xy} = \begin{cases} \left| g(x,y) - \mathrm{var}\left[R\left(g(x,y)\right) - 1 \right] \right| & , \text{if } R\left(g(x,y)\right) > R\left(\underset{S_{xy}}{MED} \right) \\ \left| g(x,y) - \mathrm{var}\left[R\left(g(x,y)\right) + 1 \right] \right| & , \text{if } R\left(g(x,y)\right) < R\left(\underset{S_{xy}}{MED} \right) \\ 0 & , \text{otherwise,} \end{cases}$$

Differential Rank Impulse Detector

- Let r be the length (in ranks) of the interval in the variational series where an impulse can be located



- Let s be the maximal acceptable difference between the intensity value of the pixel of interest and its neighbor located in the variational series between this impulse and the median. Then ● is an impulse if

d_{xy}= Var(●)-Var(●)≥ s d_{xy}= Var(●)-Var(●) ≥ s

Differential Criteria of an impulse

Differential Rank Impulse Detector

- Differential Rank Impulse Detector (DRID) works as follows
- The pixel (x,y) is considered noisy if the following condition holds:

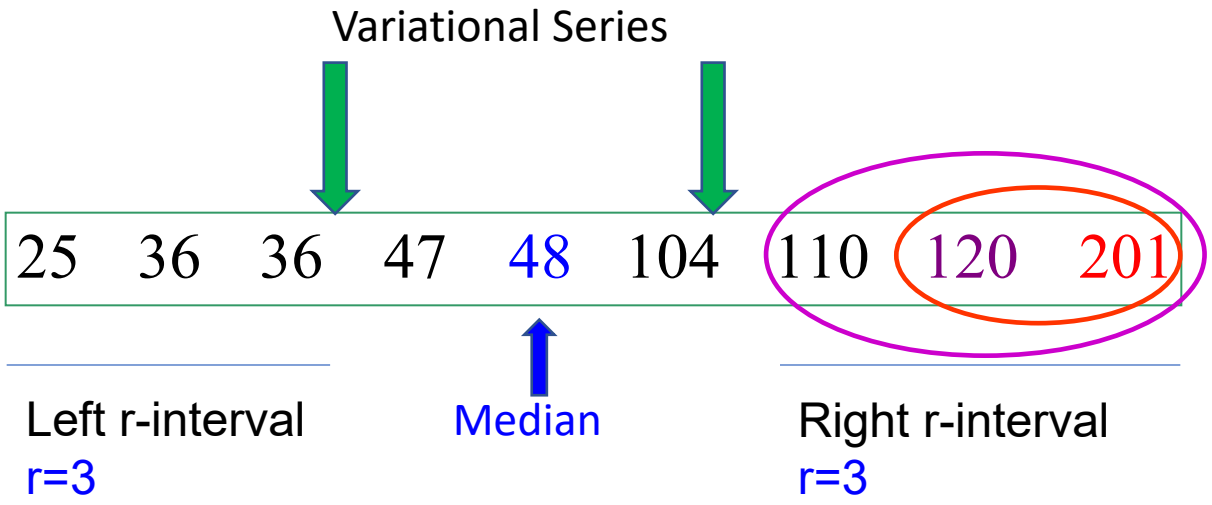
$$\left(\left(R(g(x,y)) \le r\right) \vee \left(R(g(x,y)) \ge mn - r + 1\right)\right) \wedge \left(d_{xy} \ge s\right)$$

the intensity value $g(x,y)$ in the pixel of interest is located in the r-interval from one of the ends of the variational series, and the difference d_{xy} between this value and its neighbor located in the variational series closer to the median exceeds s

Differential Rank Impulse Detector r=3, s=10

3x3 window, 201 is an intensity of interest

25	36	36
47	201	48
104	110	120



201-120 = 81>10=s → 201 is an impulse and it shall be filtered

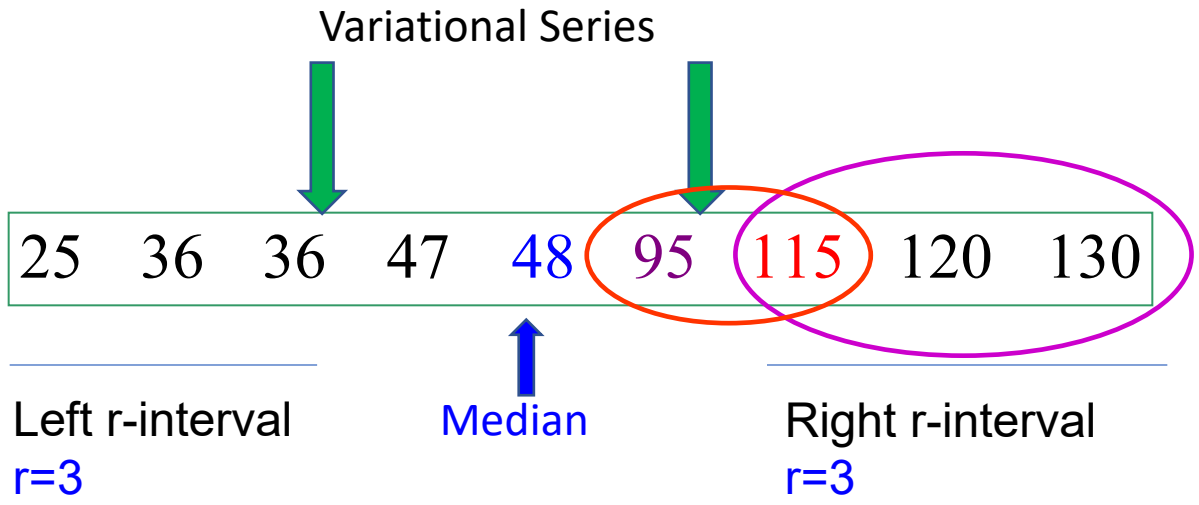
The processing result
The impulse 201 was removed

25	36	36
47	48	48
104	110	120

Differential Rank Impulse Detector r=3, s=10

3x3 window, 115 is an intensity of interest

25	36	36
47	115	48
95	120	130



115-95 = 20>10=s → 115 is an impulse and it shall be filtered

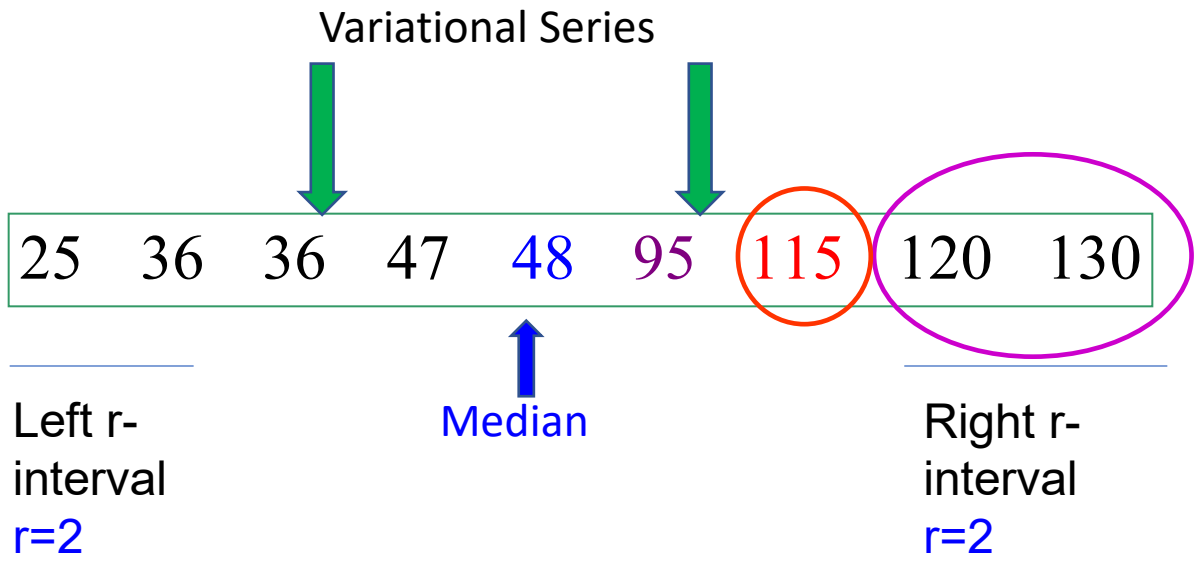
The processing result
The impulse 115 was removed

25	36	36
47	48	48
95	120	130

Differential Rank Impulse Detector r=2, s=10

3x3 window, 115 is an intensity of interest

25	36	36
47	115	48
95	120	130

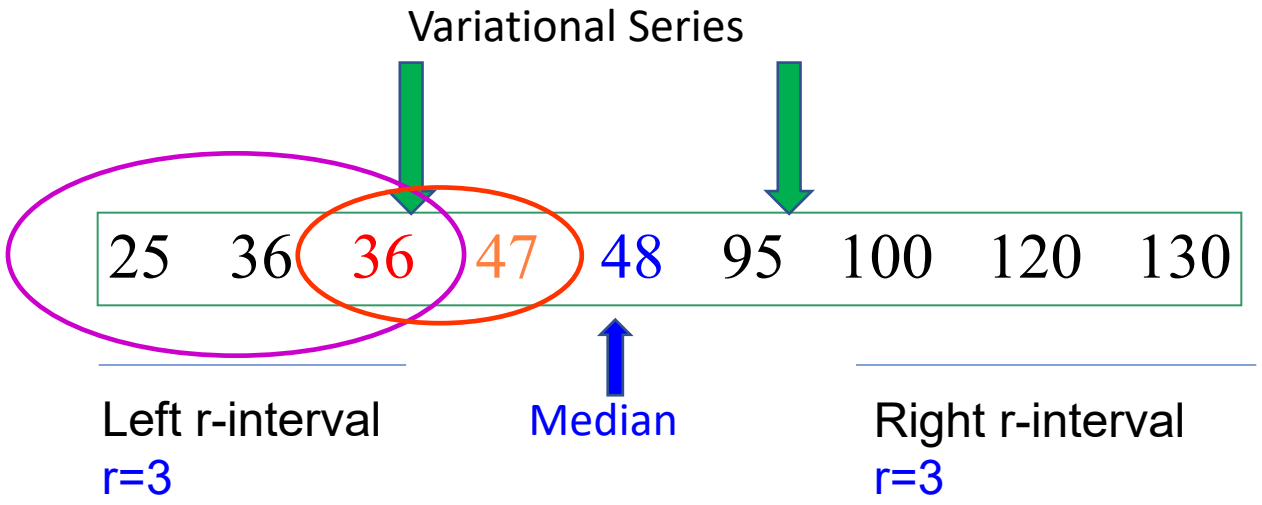


115 is **out** of both r-intervals, filtering is not needed!

Differential Rank Impulse Detector r=3, s=10

3x3 window, 36 is an intensity of interest

25	36	100
47	36	48
95	120	130



47-36 = 11>10=s → 36 is an impulse and it shall be filtered

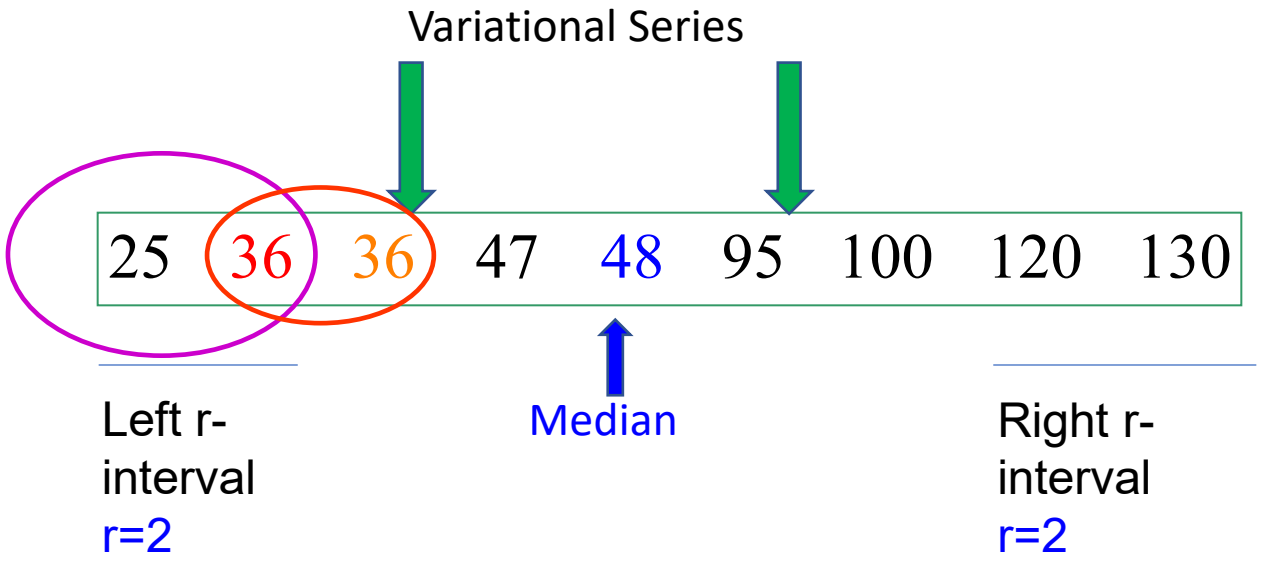
The processing result
The impulse 36 was removed

25	36	100
47	48	48
95	120	130

Differential Rank Impulse Detector r=2, s=10

3x3 window, 36 is an intensity of interest

25	36	100
47	36	48
95	120	130



36-36 = 0<10=s → 36 is not an impulse because it does not differ from its neighbor from the variational series by more than threshold s , filtering is not needed!

Differential Rank Impulse Detector: Algorithm

Determine r ("suspicious" rank interval width) and s (intensity threshold)

Take a 3x3 window around a pixel of interest g(x, y)

Create a variational series from the intensities of pixels in this 3x3 window

Evaluate rank $R\left(g(x,y)\right)$ of the intensity of interest.
If this intensity appears more than once in the variational series, its last appearance should be counted if $R\left(g(x,y)\right) \le 5$ (left r-segment) and its first appearance should be counted if $R\left(g(x,y)\right) \ge 5$ (right r-segment)

If $\left(\left(R(g(x,y) \le r\right) \vee \left(R(g(x,y)) \ge mn - r + 1\right)\right)$ is true, then a pixel of interest can be corrupted by an impulse; otherwise move to the next pixel

Evaluate

$$d_{xy} = \begin{cases} \left| g(x,y) - \mathrm{var}\left[R\left(g(x,y)\right) - 1\right] \right| & , \text{if } R\left(g(x,y)\right) > R\left(\underset{S_{xy}}{MED}\right) \\ \left| g(x,y) - \mathrm{var}\left[R\left(g(x,y)\right) + 1\right] \right| & , \text{if } R\left(g(x,y)\right) < R\left(\underset{S_{xy}}{MED}\right) \\ 0 & , \text{otherwise,} \end{cases}$$

If $\left(d_{xy} \ge s\right)$ then a pixel of interest is corrupted by an impulse – apply a median filter to filter it out; otherwise move to the next pixel

Filtering of Scratches

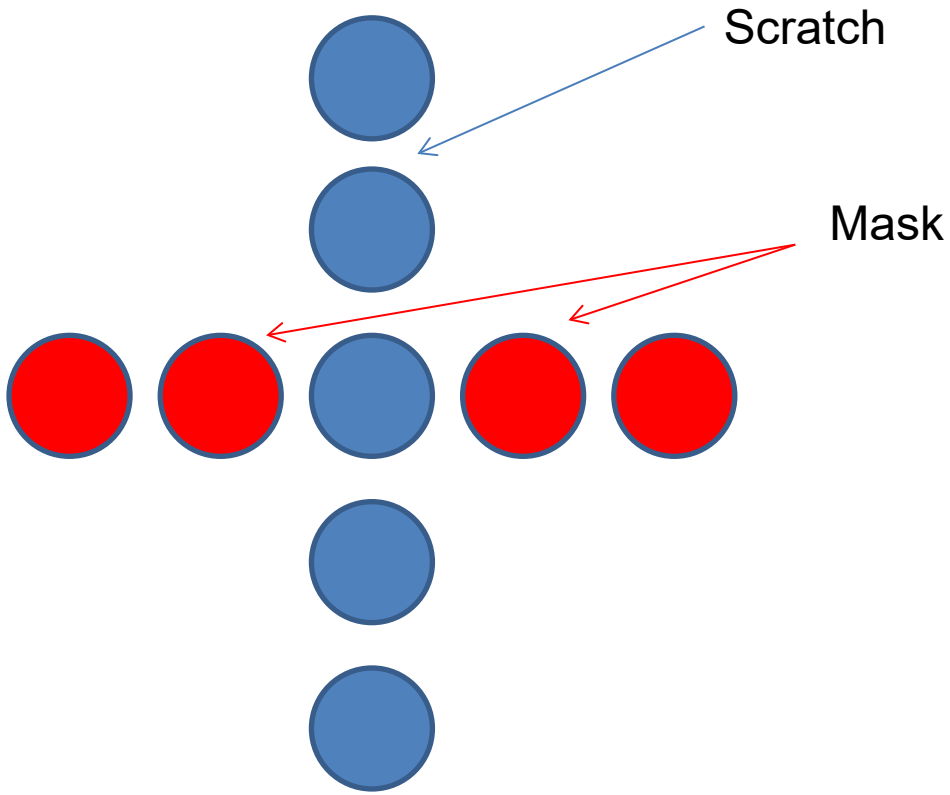
- Any scratch is a collection of impulses – there are consecutive impulses extended in some direction along some virtual curve

- To remove a scratch from an image, it is necessary to process it locally and to use the mask median filter

Filtering of Scratches

- In the mask median filter, a median is taken not over all the pixels in a filtering window, but only over those marked in the mask
- To remove a scratch, it is necessary to create a “counterscratch” mask including there pixels from the virtual perpendicular to the scratch

Filtering of Scratches



Threshold Boolean Filtering (TBF)

- Threshold Boolean filtering (also often referred to as stack filtering) is reduced to:

➢ Splitting an image with $M+1$ gray levels into M binary planes (slices) by thresholding

➢ Separate processing of the binary planes (slices) using a processing Boolean function

➢ Merging the processed binary planes into a resulting image

Threshold Boolean Filtering (TBF)

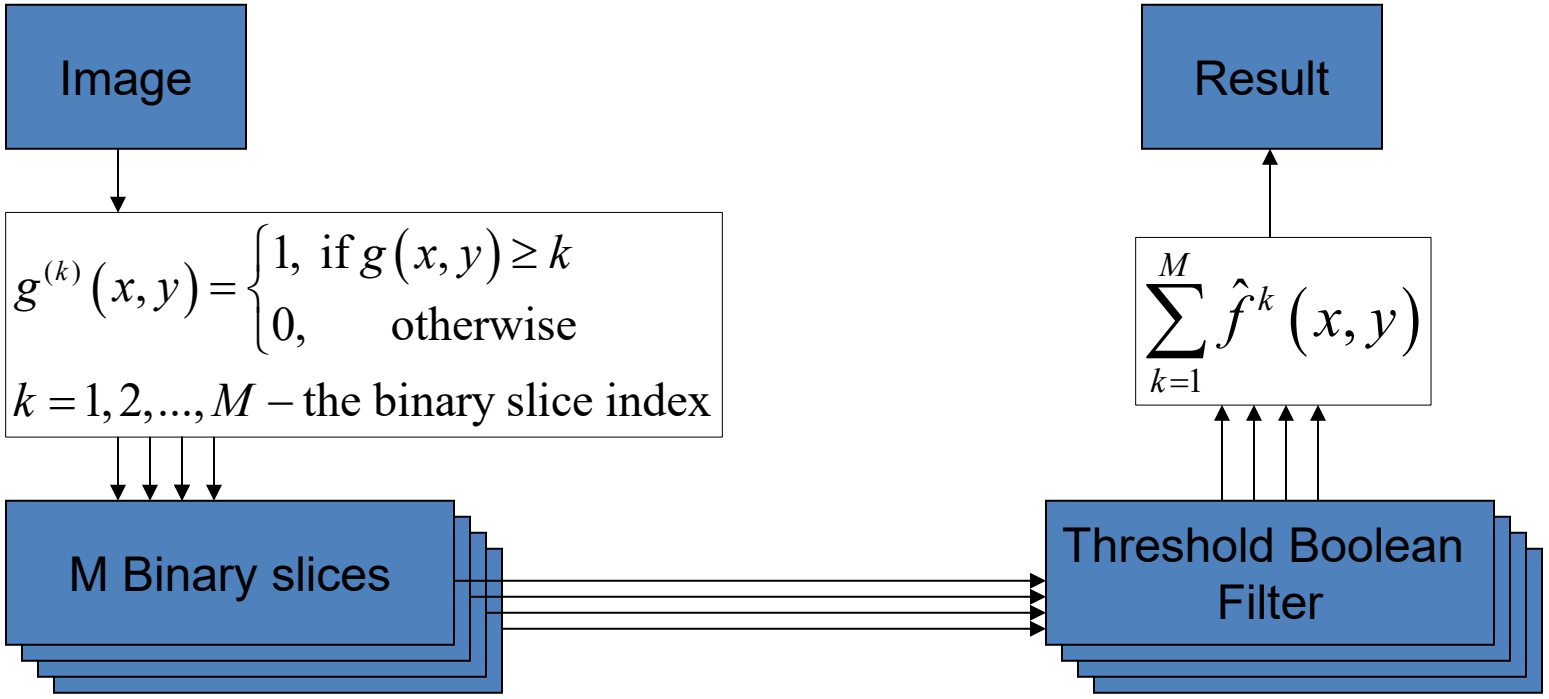
- Let g be a scale image with $(M+1)$ gray levels
- Let $g^{(k)}(x,y)=\begin{cases}1 & ,\text{ if } g(x,y)\geq k\\ 0 & ,\text{ otherwise}\end{cases};k=1,\ldots,M$ be a k^{th} binary plane
- Then a filtering operation is determined by

$$\hat{f}(x,y)=\sum_{k=1}^{M}F\left(S_{xy}^{(k)}\right)$$

where F is the processing Boolean function, $S_{ij}^{(k)}$ is a k^{th} binary plane of a local window around a pixel of interest

Threshold Boolean Filter (TBF)

The idea behind the Threshold Boolean Filter is to process each binary plane of an image separately, and after filtering all the binary planes are combined together to form the resulting image. The following diagram illustrates this idea:



Impulse Noise Filtering using TBF

The filter is defined by the following functions. For 3x3 window:

$$f(x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8,x_9) = \left[x_5 \wedge \left(\bigwedge_{j=1}^{T} \left(\bigvee_{\substack{k=i_{j_t} \\ k\neq 5}} x_k \right) \right) \right] \vee \left[\bar{x}_5 \wedge \left(\bigvee_{j=1}^{T} \left(\bigwedge_{\substack{k=i_{j_t} \\ k\neq 5}} x_k \right) \right) \right] \quad T = C_t^8$$

For 5x5 window

$$f_{5x5}(x_1,\ldots,x_{25}) = f(\ldots) \wedge \left[x_5 \wedge \left(\bigwedge_{j=1}^{P} \left(\bigvee_{\substack{k=i_{j_s} \\ k\neq 5}} x_k \right) \right) \right] \vee \left[\bar{x}_5 \wedge \left(\bigvee_{j=1}^{P} \left(\bigwedge_{\substack{k=i_{j_s} \\ k\neq 5}} x_k \right) \right) \right] \quad P = C_p^{16}$$

where X_5 is the central element of the window, i.e. the pixel under consideration,

T is the number of all possible elementary conjunctions of *t* variables out of 8 (the number of pixels in a 3x3 window),

P is the number of all possible elementary conjunctions of *s* variables out of 16 (5x5 window).

Impulse Noise Filtering using TBF

- TBF is highly efficient for filtering of random impulse noise filtering with a low corruption rate ($\leq 5\%$). It may preserve image details with a higher accuracy than other filters

- Disadvantages: strictly dependent on the successful choice of the parameters and computationally costly (compared to median and rank-order ER filters, even used in conjunction with noise detector)

Rank-Order (Order Statistic) Filters

- Rank-order filtering is based on the analysis of the variational series followed by some nonlinear averaging of a signal
- The median filter is a representative of the rank-order filters family
- We will consider rank-order filters using their classification done by Leonid Yaroslavsky

Rank-Order EV Filter

- Let ε_v be a value, which determines the following interval (**EV**) around the intensity value of the pixel of interest $g(x, y)$

$$EV = \left\{ g(i,j) : \left| g(x,y) - g(i,j) \right| \le \varepsilon_v ; g(i,j) \in S_{xy} \right\}$$

- An EV interval contains those intensity values whose difference from the intensity value in the pixel of interest does not exceed ε_v

- A suboptimal value of the parameter is $\varepsilon_v = \sigma$ - standard deviation measured over an image

Rank-Order EV Filter

- Rank-order EV filter is determined as follows

$$\hat{f}(x,y)=\begin{cases}\text{MEAN}\left(g(i,j)\right), g(i,j)\in EV \\ \text{or} \\ \text{MED}\left(g(i,j)\right), g(i,j)\in EV\end{cases}$$

- The first option leads to more careful and accurate reduction of additive noise (compared to linear filters)

Example: Rank-Order EV Filter → $\varepsilon_v = 25$

3x3 window, 110 is an intensity of interest

25	36	36
47	110	48
90	110	120

Variational Series

25 36 36 47 48 90 110 110 120

$$EV = \{90,110,110,120\}$$

$$MEAN\{S_{110}\} = 69.1 \approx 69$$

Mean over all elements in a 3x3 window

25	36	36
47	69	48
104	110	120

$$MED\{EV\} = 110$$

No changes will be made

$$MEAN\{EV\} = 107.6 \approx 108$$

25	36	36
47	108	48
104	110	120

Rank-Order EV Filter

- Rank-order EV filter preserves small details and image boundaries with a higher accuracy than linear filters. With a careful choice of ε_v it is possible to avoid smoothing of $n/2 \text{ x } m/2$ details
- Disadvantage is the dependence on the choice of the parameter ε_v.
- This filter may show better results if EV has been chosen adaptively for each S_{xy} window

Rank-Order EV Filter with mean: Algorithm

Determine ε_v It is better to choose it equal to the image standard deviation

Take a 3x3 window S_{xy} around a pixel of interest g(x, y)

Create a variational series from the intensities of pixels in this 3x3 window

Create the following set from the elements of the variational series

$$EV = \left\{ g(i,j) : \left| g(x,y) - g(i,j) \right| \le \varepsilon_v ; g(i,j) \in S_{xy} \right\}$$

Find a filtered intensity in the pixel of interest by finding mean over the EV set:

$$\hat{f}(x,y) = \mathrm{MEAN}\left(g(i,j) \right), g(i,j) \in EV$$

Rank-Order ER Filter

- Let ε_r be a value, which determines the following interval (**ER**) around the intensity value of the pixel of interest

$$ER = \left\{ g(i,j) : \left| R\left(g(x,y)\right) - R\left(g(i,j)\right) \right| \leq \varepsilon_r ; g(i,j) \in S_{xy} \right\}$$

- An ER interval contains those intensity values whose ranks differ from the rank of the intensity value in the pixel of interest by not more than ε_r

Rank-Order ER Filter

- Rank-order ER filter is determined as follows

$$\hat{f}(x,y)=\begin{cases}\text{MEAN}\left(g(i,j)\right), g(i,j)\in ER \\ \text{or} \\ \text{MED}\left(g(i,j)\right), g(i,j)\in ER\end{cases}$$

- The second option leads to more careful removal of random impulse noise with a low corruption rate (1-5%) (the filter should be applied iteratively). It can also be used to “close” impulse gaps (spots) – a window larger than a spot size should be used
- The second option is also better for speckle (multiplicative) noise

Example: Rank-Order ER Filter → $\varepsilon_r = 2$

3x3 window, 110 is an intensity of interest

25	36	36
47	110	48
90	110	120

Variational Series

25 36 36 47 48 90 110 110 120

$$ER = \{48, 90, 110, 110, 120\}$$

$$MED\{S_{110}\} = 48$$

Mean over all elements in a 3x3 window

25	36	36
47	48	48
104	110	120

$$MED\{ER\} = 110$$

No changes will be made

$$MEAN\{ER\} = 95.6 \approx 96$$

25	36	36
47	96	48
104	110	120

Rank-Order ER Filter

- Advantage of the rank-order ER filter is that it preserves small details and image boundaries with a higher accuracy than the median filter. With a careful choice of ε_r, it is possible to remove salt-and-pepper impulse noise iteratively with more careful preservation of image details
- Disadvantage is the dependence on the choice of the parameter ER.
- This filter may show better results if ER has been chosen adaptively for each S_{xy} window

Rank-Order ER Filter with median: Algorithm

Determine ε_r It should be less than a half of the number of elements in the local processing window

Take a 3x3 window S_{xy} around a pixel of interest g(x, y)

Create a variational series from the intensities of pixels in this 3x3 window

Create the following set from the elements of the variational series

$$ER = \left\{ g(i,j) : \left| R\left(g(x,y)\right) - R\left(g(i,j)\right) \right| \le \varepsilon_r ; g(i,j) \in S_{xy} \right\}$$

Find a filtered intensity in the pixel of interest by finding median over the ER set:

$$\hat{f}(x,y) = \mathrm{MED}\left(g(i,j)\right), g(i,j) \in ER$$

Rank-Order KNV Filter

- Let k be a value, which determines the following interval (**KNV**) around the intensity value of the pixel of interest

$$KNV = \left\{ \begin{array}{l} g_s = g(i,j) : \left| R(g(x,y)) - R(g(i,j)) \right| \le nm; \\ s \le k; g(i,j) \in S_{xy} \end{array} \right\}$$

- A KNV interval contains k closest intensities from S_{xy} to the intensity in xy^{th} pixel, in terms of the closeness of their intensity values to the intensity $g(x,y)$

Rank-Order KNV Filter

- Rank-order KNV filter is determined as follows

$$\hat{f}(x,y)=\begin{cases}\text{MEAN}\left(g(i,j)\right), g(i,j)\in KNV \\ \text{or} \\ \text{MED}\left(g(i,j)\right), g(i,j)\in KNV\end{cases}$$

- The first option leads to more careful reduction of additive noise (compared to linear filters)

Example: Rank-Order KNV Filter → $k = 7$

3x3 window, 110 is an intensity of interest

25	36	36
47	110	48
90	110	120

Variational Series

25 36 36 47 48 90 110 110 120

$$KNV = \{36, 37, 48, 90, 110, 110, 120\}$$

$$MED\{S_{110}\} = 48$$

Mean over all elements in a 3x3 window

25	36	36
47	48	48
104	110	120

$$MED\{KNV\} = 90$$

25	36	36
47	90	48
104	110	120

$$MEAN\{KNV\} = 95.6 \approx 96$$

25	36	36
47	96	48
104	110	120

Rank-Order KNV Filter

- Advantage of the rank-order KNV filter is that it may preserve small details whose area is > k squared pixels
- However, its ability to reduce noise is limited. It is better to use this filter only when it is absolutely necessary to preserve some small details
- Another disadvantage is its dependence on the choice of the parameter k
- This filter may show better results if k has been chosen adaptively for each S_{xy} window

Geometric Mean Filter

$$\hat{f}(x,y) = \left(\prod_{(s,t)\in S_{xy}} S(s,t) \right)^{\frac{1}{mn}}$$

- Can preserve details a little bit better than the arithmetic mean filter

Harmonic Mean Filter

$$\hat{f}(x,y) = \frac{mn}{\sum_{(s,t)\in S_{xy}} \frac{1}{g(s,t)}}$$

- Can be good for multiplicative (speckle) noise removal

Contraharmonic Mean Filter

$$\hat{f}(x,y) = \frac{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q+1}}{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q}}$$

where Q is the order of the filter

- With positive Q eliminates pepper noise , while with negative Q Eliminates salt noise
- With Q=0 it is the same as the arithmetic mean filter, and with Q=1 it is the same as the harmonic mean filter